



The BLEMS Augmented Sensor Device

ISI-TR-689

March 25, 2014

Joe Touch

touch@isi.edu

Approved for external distribution.

*This work was supported in part by:
“Building Level Energy Management Systems (BLEMS),”
Department of Energy (DE-EE0004019), 2010-2013.*

Abstract

The BLEMS Augmented Sensor Device (ASD) is a stand-alone sensor platform for gathering information on room use to support automated building HVAC tuning. The device is accessible using WiFi, and operates as a web server that returns sensor information as a coded string. This document describes the design, assembly, and programming of the ASD as currently used for the BLEMS project.

TABLE OF CONTENTS

Table of Figures: 2

1 Introduction 3

2 Design..... 4

3 Inventory 7

4 Wiring Schematic 9

5 Assembly 11

 5.1 Sensors and their connectors 11

 1.1 Assemble the shield..... 13

 1.2 Assemble the ASD box 14

6 Programming 17

7 Use..... 20

8 Acknowledgements 20

9 References 21

TABLE OF FIGURES:

Figure 1 Arduino Duemilanove (left) and Diamondback (right)..... 4

Figure 2 ASD schematic..... 9

Figure 3 Prototyping shield - fully assembled (top and bottom view), and schematic..... 10

Figure 4 ASD wiring schematic (full, left; simplified, right) 10

Figure 5 BLEMS prototyping shield 13

Figure 6 Attach 4-pin connector to pin block..... 15

Figure 7 Attach sound/temp 3-pin connector to pin block 15

Figure 8 Attach 3-pin integrated reflector sensor connector to pin block 15

1 INTRODUCTION

The BLEMS Augmented Sensor Device (ASD) is a single-board computer that measures various environment parameters and makes these values available through a simple web page accessible over WiFi, as developed for the Building Level Energy Management System (BLEMS) project [Bl] [Or11], and used in for smart buildings research [Ja13] [Ja14a] [Ja14b] [Li13] [Or14] [Ya12a] [Ya12b] [Ya13]. This document describes the design, assembly, programming, and use of the ASD.

The primary purpose of the ASD is to provide fine-grained remote measurement of environment parameters that might be useful in predicting room occupancy and usage in an office building. The BLEMS project uses derived information of room occupancy patterns to predict future usage and to optimize building power and HVAC systems. The ASD device includes a number of sensors as a preliminary experiment, to help determine which sensors correlate most closely to actual room usage. The ASD reporting information is intended to be correlated with “ground truth”, self-reported room occupancy logs voluntarily maintained by room occupants.

The set of included sensors was based on a combination of those economically available, those perceived as likely to correlate to room usage and activity, and that can be easily integrated into an inexpensive platform that can be deployed throughout a medium-sized campus building at a university. The platform supporting the sensors needed to support remote data collection and monitoring to facilitate automated data collection.

2 DESIGN

The ASD consists of an Arduino Diamondback or BlackWidow¹ stand-alone single-board microcontroller computer based on the Arduino Duemilanove with integrated 802.11 WiFi, with sensor devices to support the BLEMS project (Figure 1).



Figure 1 Arduino Duemilanove (left) and Diamondback (right)

The Diamondback and BlackWidow are equivalent configurations with the following properties:

- 16Mhz 8-bit CPU
- 32KB flash RAM (2KB used for the bootloader), as program memory
- 2KB SRAM, as data memory
- 1KB EEPROM, used for per-device info, such as the MAC address
- 802.11b WIFI 1-2 Mbps with WEP, WPA, WPA2 (infrastructure and ad-hoc)
- 6 analog input pins (all available to the user)
- 14 digital I/O pins (the USB uses pins 0,1, and WiFi uses 2 for interrupt, 9 for the status LED, 10, 11, 12, 13 for the SPI bus, leaving pins 3-8 for the user)

The BLEMS project includes the following sensors, which provide an output signal as indicated:

- Ambient light – average diffuse visible light level, an analog sensor that outputs 0-5V corresponding to the visible light sensed.



- Sound – average sound, an analog sensor that outputs 0-5V corresponding to the air pressure sensed (a pure tone would output a sinusoid waveform).



¹ The BlackWidow was discontinued Summer 2011, and was replaced by the Diamondback, which is both hardware and software equivalent.

- CO₂ level – carbon dioxide level, a self-calibrated digital sensor that outputs positive pulses (start at rising transition, end at falling transition) whose duration (in seconds) corresponds to the level of carbon dioxide, according to a table.



- Motion – motion in a broad field, a digital sensor that outputs a high (digital '1') when motion is detected within a beam of around 30 degrees. The sensor is triggered by motion, and reset only in the absence of motion for over approximately 10 seconds.



- Temperature and humidity – air temperature and humidity, a digital sensor with a particular protocol as follows:



The single data pin is bidirectional as follows:

- Pull low for 2ms
- Pull high for 10ms
- Read the data, which arrives as positive pulses (start at low-high transition, end at high-low transition); pulses shorter than 40ms are 0's, and longer (though less than 150ms) are 1's
- The data arrives MSB-LSB (most significant bit first) order, as 5 bytes:
 - Humidity high order byte (HHB) - integer part
 - Humidity low order byte (HLB) - decimal part
 - Temperature high order byte (THB) - integer part
 - Temperature low order byte (TLB) - decimal part
 - Checksum byte (CB)
- The data obeys the following checksum:
 - $CB = HHB + HLB + THB + TLB$
- The temperature and humidity are each a pair of bytes representing the integer and decimal parts of the values representing:
 - Humidity is percentage of relative humidity, i.e., 85 means 85% (or 0.85) relative humidity (uncalibrated), accurate to +/-2%
 - Temperature is a count of hundredths of degrees Celsius, i.e., 5583 means 55.83° C (uncalibrated), accurate to +/-0.5°C

- Door switch (magnetic reed switch) – door open/closed, a normally open 2-contact switch which is closed in the presence of a magnet. It is used as a digital sensor.



- Object presence – presence of an object in a narrow field, a digital sensor that emits a light beam and detects its reflectance within a narrow beam of around 15 degrees, and a distance of around 3 feet. It outputs a 1 when an object is present and a 0 otherwise.



The BLEMS project measures parameters of direct interest, including temperature, humidity, and ambient light. It also measures parameters of potential interest, including door open/closed status, and carbon dioxide level as rough measure of air quality. All other sensors (motion, object presence, sound) are intended to allow users to estimate the number of room occupants and their activity level; for this purpose, the carbon dioxide level may also be helpful. The motion sensor, sound, and carbon dioxide levels together indicate occupant activity. The carbon dioxide level and object presence may help estimate the room occupancy. Only the carbon dioxide sensor is pre-calibrated; all other sensor values should be considered relative values only.

3 INVENTORY

The following is the list of parts required for a single BLEMS sensor, with current approximate costs (excluding tax and shipping) and the estimated total to build 60 boxes:

Description	Source	No.	Unit Price	Total Price
Microcontroller and Enclosure Components				
Arduino Diamondback Microcontroller	http://www.cutedigi.com/product_info.php?products_id=4570	60	\$75	\$4500
Prototyping shield	http://www.makershed.com/ProductDetails.asp?ProductCode=MKAD6	60	\$15	\$900
USB Cables	http://www.amazon.com/Cables-Go-27005-MiniBCable/dp/B000067RVL/ref=sr_1_4?s=electronics&ie=UTF8&qid=1295644752&sr=1-4	5	\$3	\$15
Wall brick power supply	http://www.makershed.com/ProductDetails.asp?ProductCode=MKSF3	60	\$7	\$420
4-pin 3-wire 2mm male connector (sound, digital temp/humidity)	http://www.seeedstudio.com/depot/electronic-brick-3-pin-to-grove-4-pin-converter-cable-5-pcs-pack-p-728.html?cPath=178_179	24 packs (5/pack)	\$4.50	\$108
4-pin 4-wire 2.5mm male connector (motion, light, CO2, jumper)	http://www.seeedstudio.com/depot/4-pin-dualfemale-jumper-wire-300mm-5-pcs-pack-p-97.html?cPath=175_187	24 packs (5/pack)	\$3	\$72
4-pin 2.5mm JST connector (CO2)	http://www.batteryspace.com/connectoradaptor4pinfemalejstxhconnector.aspx	60	\$2	\$120
M Coax Panel Power Mount	http://www.radioshack.com/product/index.jsp?productId=2102486	60	\$3	\$180
Black Project Enclosure Box	http://www.mpja.com/prodinfo.asp?number=15888+BX	60	\$6.50	\$390
Power Socket	http://www.robotshop.com/seeedstudio-jst-2-pin-power-connector.html?utm_source=google&utm_medium=base&utm_campaign=jos	60	\$1.10	\$66
2-conductor jack/socket for reed switch	http://www.radioshack.com/product/index.jsp?productId=2103452	30 packs (2/pack)	\$3	\$90
Header pins	http://www.robotshop.com/sfe-straight-pin-headers.html	60	\$2.25	\$135
Precut wires, 2in length	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=923345-20-ND	4 (100/pack)	\$13.72	\$55

Sensor Components				
Photoreflective	http://www.seeedstudio.com/depot/electronic-brick-diffuse-reflection-ir-switch-sensor-analog-p-595.html?cPath=144_149	60	\$11	\$660
Magnetic reed switch	http://www.seeedstudio.com/depot/magnetic-door-switch-p-666.html?cPath=156_160	60	\$2	\$120
Motion	http://www.cutedigi.com/product_info.php?cPath=278_295&products_id=4290&osCsid=bc9f944c39dc1c44156dc4381ad0bbbe	60	\$7	\$420
CO2	http://www.co2meter.com/collections/co2-sensors/products/k-22-pwm-co2-sensor	30	\$60	\$1800
Temperature and humidity	http://www.seeedstudio.com/depot/Grove-TemperatureHumidity-Sensor-Pro-p-838.html?cPath=25_125 http://www.seeedstudio.com/depot/twig-temphumi-sensor-p-745.html?cPath=144_147	60	\$15	\$900
Ambient light	http://www.liquidware.com/shop/show/SEN-LTE/AMBI+Light+Sensor	60	\$5	\$300
Sound	http://www.seeedstudio.com/depot/twig-sound-sensor-p-752.html?cPath=144_148	60	\$5	\$300
Estimated 60-box Total (\$193/box):				\$11551

Other incidental parts:

- Solder
- Heat-shrink tubing
- Hot-melt glue

Equipment:

- Soldering iron (low wattage, with a grounded tip)
- Hot-melt glue gun
- Wire cutters/strippers
- Volt/ohm meter
- Oscilloscope (for debugging)

4 WIRING SCHEMATIC

The shield signal pins are connected to the sensors as shown in the schematic in Figure 2. Unshaded signals are available for use.

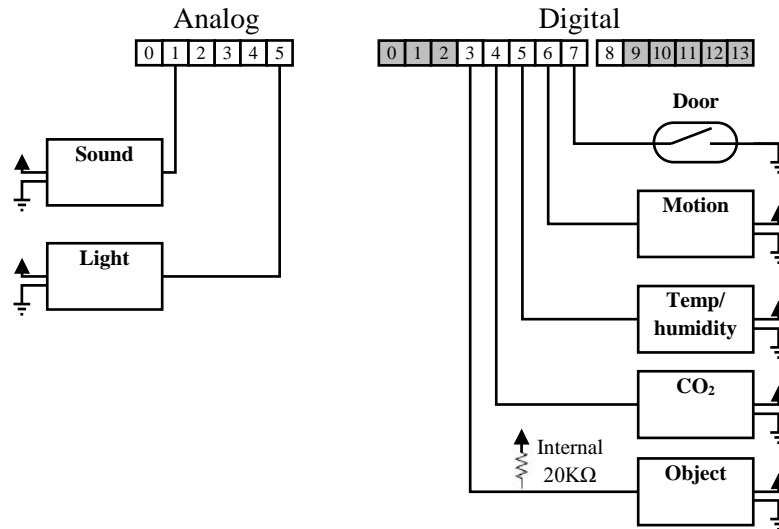


Figure 2 ASD schematic

In the ASD, the signal pins are connected as follows (available pins shown in bold):

- Digital pin 0 = *not available (USB)*
- Digital pin 1 = *not available (USB)*
- Digital pin 2 = *not available (interrupt for WiFi)*
- **Digital pin 3 = Object presence**
- **Digital pin 4 = CO₂ level**
- **Digital pin 5 = Temperature and humidity**
- **Digital pin 6 = Motion**
- **Digital pin 7 = Door switch**
- Digital pin 8 = *not used (available for power control)*
- Digital pin 9 = *not available (WiFi status)*
- Digital pin 10 = *not available (SPI)*
- Digital pin 11 = *not available (SPI)*
- Digital pin 12 = *not available (SPI)*
- Digital pin 13 = *not available (SPI)*
- **Analog pin 0** = *not used*
- **Analog pin 1 = Sound**
- **Analog pin 2** = *not used*
- **Analog pin 3** = *not used*
- **Analog pin 4** = *not used*
- **Analog pin 5 = Ambient light**
- The Arduino Diamondback uses the layout of the Arduino Duemilanove and supports a common development daughtercard, known as a “shield” (Figure 3).

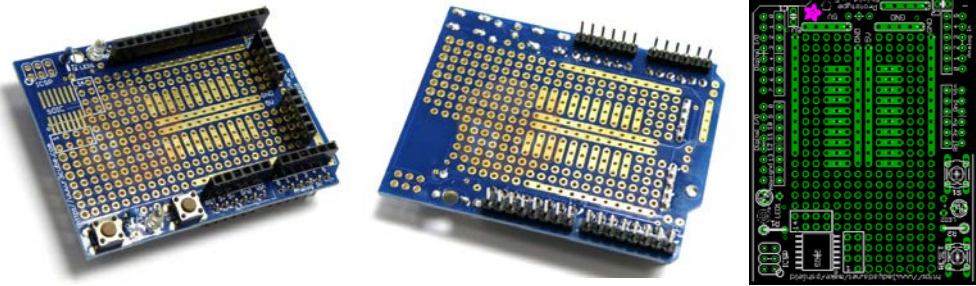


Figure 3 Prototyping shield - fully assembled (top and bottom view), and schematic

The BLEMS configuration of the shield uses These wires are implemented on the Ardunio shield as shown in Figure 4, left. To simplify assembly, the blue wires in the left diagram are replaced with a single 4-wire jumper, using male headers signals (as shown on the right diagram).

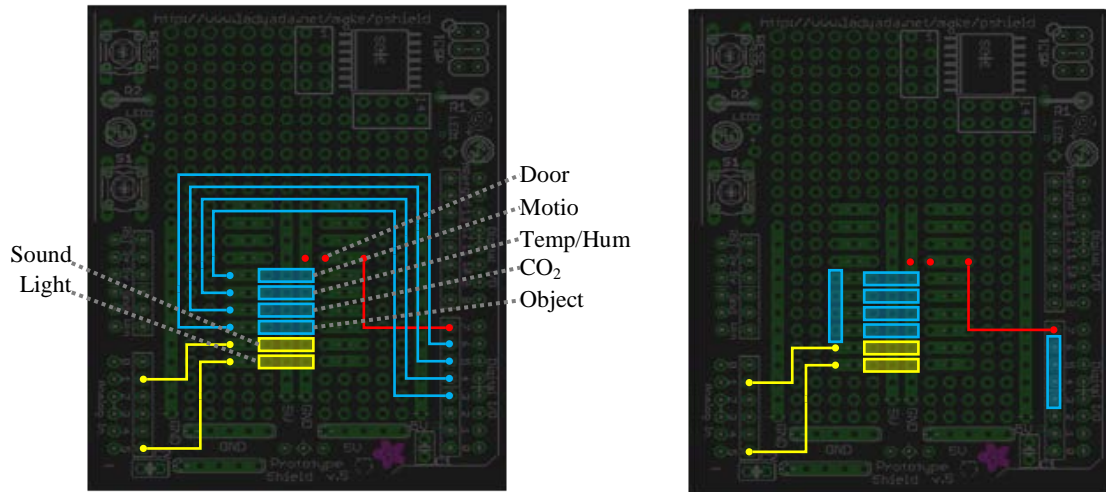


Figure 4 ASD wiring schematic (full, left; simplified, right)

5 ASSEMBLY

The ASD is assembled in stages as follows.

5.1 SENSORS AND THEIR CONNECTORS

ASD fabrication follows the following steps. Note that this document does NOT YET address how the resulting system is installed in a box.

- 1) Add connectors to the sensors (the object sensor already has the necessary connector):

- Light – insert 2.5mm 4-wire connector onto the sensor



NOTE – connect as follows:

Sensor GND to 4-wire red-wire side

- Sound – insert 2mm connector onto the sensor (one-way connector)



- Temp/humidity – insert 2mm connector onto the sensor (one-way connector)



- Motion – solder one half of a 2.5mm 4-wire connector onto the sensor



NOTE – connect as follows:

Sensor yellow wire to 4-pin yellow wire

Sensor black wire to 4-pin red wire

Sensor red wire to 4-pin black wire

- CO2 – insert 2.5mm JST connector onto the sensor (one-way connector), then solder the other half of a 2.5mm 4-wire connector onto the sensor



NOTE – connect as follows:

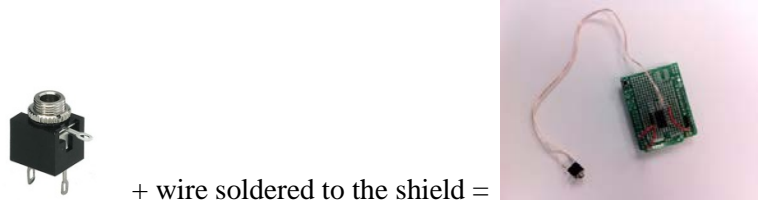
Sensor black wire to 4-pin black wire
 Sensor yellow wire to 4-pin red wire
 Sensor blue wire to 4-pin yellow wire

2) Solder the jacks and door switch

- Door switch – solder to a length of wire (12’); solder the wire to a 1/8” plug



- Door jack – solder a length of wire (8”) to a 1/8” jack; solder the wire onto the two door pads on the proto board as indicated in Figure 4, right



- Power jack – solder the power tail to the power jack



1.1 ASSEMBLE THE SHIELD.

The following steps explain how to wire the ASD shield and attach the wired sensors:

- 1) The BLEMS ASD does not use the fully populated prototyping shield (Figure 5). It uses only the following shield parts, and the remaining parts are not used (note that board color varies):
 - Bottom header pins, as shown Figure 5, left image. (this is easier if connectors are inserted in a Diamondback multicontroller during assembly)
 - Top reset button, as shown in Figure 5, bottom left of right image
 - Top power capacitors, as shown in Figure 5, right side of right image

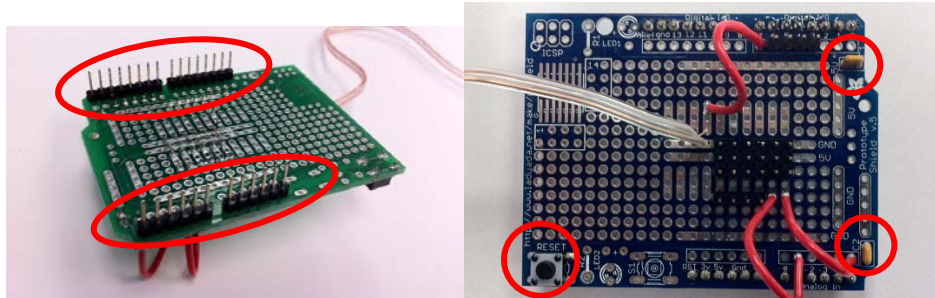
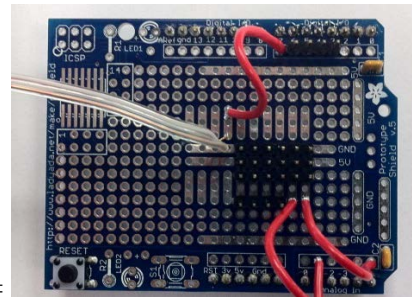


Figure 5 BLEMS prototyping shield

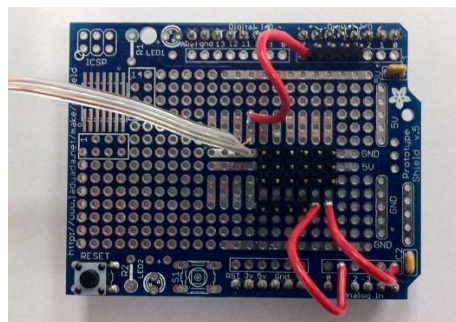
- 2) Add connectors to the shield as follows (boxes are groups of male header pins on the topside; lines are wires). Yellow indicates analog signals. Red indicates the door switch. Blue indicates other digital signals.
 - Add the headers where the boxes in Figure 4 indicate. Note that the block of 3x6 pins may be easier to install as 3 rows of 6 pins than 6 rows of 3.



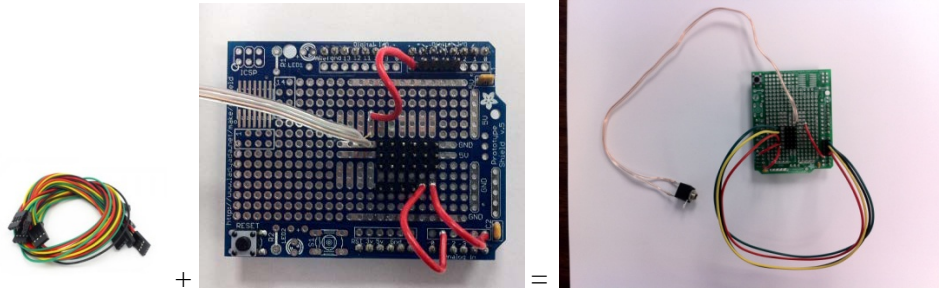
in boxes of Figure 4, right =



- Add three wires as indicated by the lines in Figure 4, right.



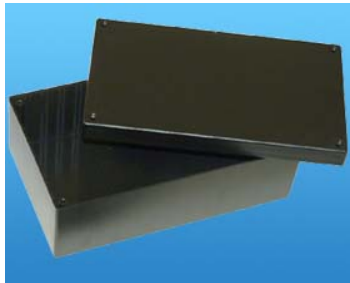
- Connect the two blue jumpers in Figure 4, right, with a 2.5mm 4-wire connector – make sure to correlate to the wiring in Figure 2



1.2 ASSEMBLE THE ASD BOX

The following steps are used to assemble the Arduino Diamondback, the ASD shield, and the sensors in a single BLEMS ASD box:

- 1) Place the devices in the box
 - Drill holes as needed



- For through-devices:
 - Power jack 7/16"
 - Door jack 1/4"
 - Reflector counter 3/4"
 - Mini-USB access port 1/2"
- For 'behind' devices
 - Motion 3/4" or 7/8"
 - Light 3/16" or 1/4"
 - Sound 1/4"
 - Temp/humidity 3/4"
- Connect the power jack to the microcontroller
- Secure screwed devices to corresponding holes in the case
 - Power jack
 - Door jack
 - Object presence sensor
- Secure the microcontroller with proto daughtercard attached inside the case
 - Also secure the antenna (on the Diamondback it's integrated and not movable, but on the discontinued Blackwidow it's a wire) so it is likely to face the center of the building when the box is mounted, and away from other sensors and the microcontroller

- Connect the following sensors via their female plugs onto the male headers at the 3x6 pin block. Note that there is the “ground side” of the block, a “signal side” of the block (nearer the 4-pin block for the jumper wire), and a middle “5v row” of the block. The following are connected in-order, from the side of the block closer to the capacitors, as indicated in Figure 5, right:
 - Light – red-wire side of the connector to the block ground-side (the green-wire side of the connector does not have a corresponding pin)

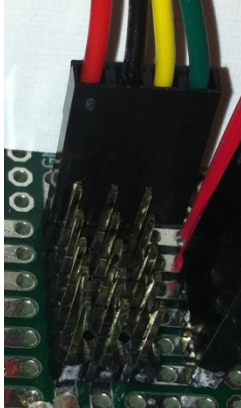


Figure 6 Attach 4-pin connector to pin block

- Sound – black-wire side of the connector the block ground-side

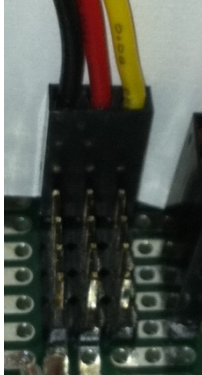


Figure 7 Attach sound/temp 3-pin connector to pin block

- Object presence – green-wire side of the connector the block ground-side

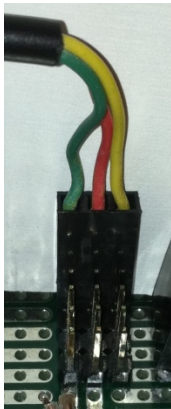


Figure 8 Attach 3-pin integrated reflector sensor connector to pin block

- CO2 – red-wire side of the connector to the block ground-side (the green-wire side of the connector does not have a corresponding pin), as shown in Figure 6
 - Temp/humidity – black-wire side of the connector the block ground-side, as shown in Figure 7
 - Motion – red-wire side of the connector to the block ground-side (the green-wire side of the connector does not have a corresponding pin) , as shown in Figure 6
 - Secure the following sensors behind their case holes:
 - Light
 - Motion
 - Secure the remaining sensors inside, in an area with reasonable air flow:
 - Sound
 - Temp/humidity
 - CO2
- 2) Prepare the box for use
- Connect the 110v/9v power transformer brick plug to the power jack
 - Connect the door switch plug to the door jack
 - Mount the box so the object presence sensor is within 3' of the main room door opening, approximately at head/body level and so its beam crosses the path of occupants entering/leaving
 - Mount the door switch near the door jamb, with the magnet on the door, so the two devices are within 1/8" of each other and aligned when the door is fully closed

6 PROGRAMMING

First install the Arduino software

URL: <http://www.arduino.cc/> (click where it says “Downloads” on that page)

Release: 1.0

Config. info. Set for the Duemilanove processor

Follow the instructions there for your platform (PC, Mac, Linux).

Then install the DiamondBack software stack, which includes drivers for the WIFI 802.11, as well as demo programs:

URL: https://github.com/CapnBry/WiShield_user_contrib (click “Downloads”)

Originally from: <https://github.com/asynclabs/WiShield>

Release: 1.3.0 (updated to new compiler, etc.)

Config. info. Unpack and place under <program-home>/libraries/WiShield/

On a PC, this is in the folder you unpacked from Arduino, which includes the Arduino application.

On a MAC, this is in the hidden folder associated with the Arduino application.

On Linux, this is wherever you unpack the Arduino software.

You should install the drivers if it asks. On a MAC, you may also need to install the USB drivers here:

URL: <http://www.ftdichip.com/Drivers/VCP.htm>

Release: Mac OS-X x64 drivers

Note: DO NOT use the system CC compiler. DO NOT modify ANY of the Arduino files. The Arduino system comes with its own embedded cross-compiler. It’s designed to be installed and used ONLY through the Arduino main program interface.

In the WiShield folder, there is a file called “apps-conf.h”. Make sure “#define APP_WISERVER” is the only member of the group that is uncommented (the list of other #defines may change depending on release):

Desired result:

```
//#define APP_WEBSERVER
//#define APP_WEBCLIENT
//#define APP_SOCKAPP
//#define APP_UDPAPP
#define APP_WISERVER
```

The following is the header of the ASD software. The software should be placed in a file named “fullserver.pde”. Place that in a folder named “fullserver” in the Arduino folder in your user document area. Open the source code using the Arduino software programming environment (from the File menu). The first time you run the program, don’t forget to select the proper board type (Arduino Duemileno v2 w/ ATmega328P).

```
// -----
//
//          BLEMS SENSOR DEVICE
//
//      USC Information Sciences Institute (USC/ISI)
//          Marina del Rey, California 90292, USA
//          Copyright (c) 2011
// -----
//
// Copyright (c) 2011-2012 by the University of Southern California.
// All rights reserved.
//
// Permission to use, copy, modify, and distribute this software and its
// documentation in source and binary forms for non-commercial purposes and
// without fee is hereby granted, provided that the above copyright notice
// appear in all copies and that both the copyright notice and this permission
// notice appear in supporting documentation, and that any documentation,
// advertising materials, and other materials related to such distribution and
// use acknowledge that the software was developed by the University of
// Southern California, Information Sciences Institute. The name of the
// University may not be used to endorse or promote products derived from this
// software without specific prior written permission.
//
// THE UNIVERSITY OF SOUTHERN CALIFORNIA MAKES NO REPRESENTATIONS ABOUT THE
// SUITABILITY OF THIS SOFTWARE FOR ANY PURPOSE. THIS SOFTWARE IS PROVIDED "AS
// IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT
// LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
// PARTICULAR PURPOSE.
//
// Other copyrights might apply to parts of this software and are so noted when
// applicable.
//
// -----
// Author: Joe Touch / touch@isi.edu
//
// Based on Arduino sample code "simpleserver"
//
// VERSION HISTORY:
// 0.5 - 3/22/11 - first version tested in RGL
//      added reflector counter
// 0.62 - 5/20/11
//      added sound averaging
```

```

//      ceased reporting analog RF, temp, humidity
// 0.63 - 6/2/11
//      added support for KHALL
// 0.64 - 7/19/11
//      added motion counter
//      removed unused code for analog RF, temp, humidity, and power control
// 0.65 - 7/20/11
//      added longsound totals and averaging
// 0.66 - 8/15/11
//      added doorcount
// 0.67 - 9/6/11
// restored HIGH to door pin to set internal pullup
// 0.68 - 9/7/11
//      added door count debounce
// 0.69 - 12/6/11
//      removed MAC code, removed debug to serial
//      added added HOMENAT and JTWIFI2
//      moved DOOR to pin 7 from pin 2 (requires box rewire)
// 0.70 - 12/14/11
//      recompiled under Arduino 1.0
// 0.71 - 12/16/11
//      using updated, more robust server from shared web libraries
//      using precomputed WEP key to join network quicker
// 0.72 - 12/19/11
//      extended time printout to include days, printout strings via F() as program data
// 0.73 - 1/17/12
//      modified time printout to avoid overflow showing as negatives
//      added support for software reset, including commands to set reset timeout from
the web
// 0.74 - 4/17/12
//      added pin 8 to RST line to force hardware reset upon timeout
// 0.75 - 8/15/12
//      corrected temp and humidity output
// 0.76 - 8/15/12
//      fixed temp conversion code bug
// 0.77 - 8/15/12
//      revise temp/humidity conversion to code provided by chip manufacturer, as well as BCD
version
// 0.78 - 8/16/12
//      revise temp/humidity based on spec sheet http://www.micropik.com/PDF/dht11.pdf
//      remove BCD version; included DHT22 version
// 0.79 - 8/21/12
//      fix float calculation of temp/humidity

```

7 USE

The board can be accessed through a URL using its IP address, e.g.:

```
http://10.0.0.1/
```

The response will be a string as follows (wrapped for convenience, but printed as a single line):

```
v0.7.9 6d11:02:22 WTO 60 PTO 0 AS 5 ASAVG 55 LASTOT 64819 LASTNUM 10000 LASAVG 3 AL 2  
BM 0 BMC 2 BR 1 BRC 8 BD 1 BDC 3 COTWO 0.22 DH 81 DT 21 DH 22 DH22 81.92 DT 21.55
```

The meaning of this string is:

- v0.7.9 – version number
- 6d11:02:22 – 6 days, 11 hours, 2 minutes, 22 seconds since last reboot
- WTO – web timeout, in seconds (0 = no timeout); restart if not contacted over the web (via HTTP request) at least once every WTO seconds
- PTO – periodic timeout, in seconds (0 = no timeout); restart every PTO seconds
- AS 5 – instantaneous analog sound value (0-1023, 0 = silence), uncalibrated
- ASAVG 55 – analog sound average, as a weighted decreasing average
- LASTOT 64819 – total of the last LASNUM sound samples
- LASNUM 10000 – number of sounds samples in each LASTOT sum
- LASAVG 3 – weighted decreasing average of LASTOT values
- AL 2 – analog light (0-1023, 0 = dark), uncalibrated
- BM 0 – binary motion (1 = motion detected)
- BMC 2 – binary motion count (number of times BM has changed since reboot)
- BR 1 – binary reflection of object (1 = object present), an instantaneous reading
- BRC 8 – binary reflection count (number of times BR has changed since reboot)
- BD 1 – binary door switch (1 = door open)
- BDC 3 – binary door count (number of times BD has changed since reboot)
- COTWO 0.22 – carbon dioxide level (0-1.0, see device spec for interpretation)
- DH 81 – digital humidity level (percent relative humidity, uncalibrated)
- DT 21 – digital temperature (degrees Celsius, uncalibrated)
- DH22 81.92 – digital humidity level percent relative humidity, uncalibrated)
- DT22 21.55 – digital temperature (degrees Celsius, uncalibrated)

Only one of DH/DT and DH22/DT22 readings will be valid; the former is for the low-precision device (+/-2°C), and the latter the high-precision device (+/-0.5°C). The high-precision device can also measure temperatures below 0°C.

Note: do NOT take readings more rapidly than once every 10 seconds (preferably once every 30). The device is single-threaded and has very little buffering, and the software has been known to lockup due to overlapping pending requests.

8 ACKNOWLEDGEMENTS

This work was supported in part by: “Building Level Energy Management Systems (BLEMS),” Department of Energy (DE-EE0004019), 2010-2013.

9 REFERENCES

- [Bl] BLEMS website, <http://i-lab.usc.edu/blems.html>
- [Ja13] Jazizadeh, F., Ghahramani, A., Becerik-Gerber, B., Kichkaylo, T., and Orosz, M., “Personalized Thermal Comfort Driven Control in HVAC Operated Office Buildings,” to be presented at the 2013 ASCE International Workshop on Computing in Civil Engineering (IWCCE) Conference, June 2013.
- [Ja14a] Jazizadeh, F., Ghahramani, A., Becerik-Gerber, B., Kichkaylo, T., and Orosz, M., “A Human-Building Interaction Framework for Personalized Thermal Comfort Driven Systems in Office Buildings,” ASCE Journal of Computing in Civil Engineering, January/February 2014, in press.
- [Ja14b] Jazizadeh, F., Ghahramani, A., Becerik-Gerber, B., Kichkaylo, T., Orosz, M., “User-Led Decentralized Thermal Comfort Driven HVAC Operations for Improved Efficiency in Office Buildings,” <http://dx.doi.org/10.1016/j.enbuild.2013.11.066>, Energy and Buildings, 70 (2014), February 2014, pp 398-410.
- [Li13] Li, N, Yang, Z., Becerik-Gerber, B. and Orosz, M., “Towards Energy Savings from a Bimodal Occupancy Driven HVAC Controller in Practice,” presented at the 30th CIB W78 International Conference 2013, Beijing, China, 9-12 October 2013.
- [Or11] M. Orosz, P. Marshall, G. Roesler, T. Kichkaylo, T. Ryutov, J. Touch, B. Becerik-Gerber, C. Fern, “[Behavior Driven Building Energy Management System](#),” SERDP/ESTCP Partners in Environmental Technology Technical Symposium & Workshop poster, Nov. 2011.
- [Or14] Orosz, M., Jazizadeh, F., Becerik, B., “Building Level Energy Management Systems (BLEMS),” presented at the 2014 ARPA-E Energy Innovation Summit Technology Showcase, 24-26 February, 2014, National Harbor, MD.
- [Ya12a] Yang Z, Li N, Becerik-Gerber B, and Orosz M. “A Non-Intrusive Occupancy Monitoring System for Demand Driven HVAC Operations,” (<http://cedb.asce.org/cgi/WWWdisplay.cgi?289937>), Proceedings, CRC 2012: Construction Research Congress: Construction Challenges in a Flat World, ASCE, Reston, VA 21-22 May 2012, 828-837
- [Ya12b] Yang Z, Li N, Becerik-Gerber B, Orosz M. “A Multi-Sensor Based Occupancy Estimation Model for Supporting Demand Driven HVAC Operations,” in Proceedings of Simulation for Architecture and Urban Design (SimAUD) (<http://toc.proceedings.com/14527webtoc.pdf>), Orlando, FL, 26-30 March 2012, pp. 100-107
- [Ya13] Yang, Z., Li, N., Becerik-Gerber, B., and Orosz, M., “A Systematic Approach to Occupancy Modeling in Ambient Sensor Rich Office Environments,” Simulation: Transactions of the Society for Modeling and Simulation International, March 2013, in press.